

**IN THE CLAIMS**

Please add the following new claims.

1. (Previously Presented) A method comprising:  
maintaining a log of a plurality of requests in a storage server, each of the requests corresponding to a write operation to be performed by the storage server on a set of storage devices, the log including a separate log entry for each of the requests; and  
including a separate checksum in each of the log entries, each checksum for use by a checksum algorithm in determining data integrity of the corresponding log entry.
2. (Original) A method as recited in claim 1, wherein the requests originate from a set of client devices serviced by the storage server.
3. (Original) A method as recited in claim 1, further comprising selecting the checksum algorithm based on a desired balance between performance and checksum strength.
4. (Original) A method as recited in claim 1, further comprising automatically selecting the checksum algorithm based on a predetermined criterion.
5. (Original) A method as recited in claim 4, further comprising including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms, wherein said automatically selecting the

checksum algorithm comprises selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

6. (Original) A method as recited in claim 1, further comprising:  
 including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms; and  
 automatically selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

7. (Original) A method as recited in claim 1, further comprising including a separate algorithm variable in each of the log entries, to specify a checksum algorithm to be used separately for each said log entry.

8. (Original) A method as recited in claim 1, further comprising:  
 maintaining an entry count in the log to indicate the number of log entries in the log; and  
 using the checksum of one of the log entries to determine whether the entry count is corrupted.

9 – 23 (Canceled)

24. (Previously Presented) A storage server comprising:  
 means for receiving a plurality of requests from a set of client devices, each request corresponding to a operation to be performed by the storage server in relation to a set of storage devices; and  
 means for maintaining a log of the requests in the storage server, the log including a separate log entry for each of the requests, the log further including a

separate checksum in each of the log entries, each checksum for use by a checksum algorithm in determining data integrity of the corresponding log entry.

25. (Original) A storage server as recited in claim 24, further comprising means for selecting the checksum algorithm based on a desired balance between performance and checksum strength.

26. (Original) A storage server as recited in claim 24, further comprising means for automatically selecting the checksum algorithm based on a predetermined criterion.

27. (Original) A storage server as recited in claim 26, further comprising means for including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms, wherein said means for automatically selecting the checksum algorithm comprises means for selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

28. (Original) A storage server as recited in claim 24, further comprising:  
an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms; and  
means for automatically selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

29. (Original) A storage server as recited in claim 24, further comprising means for including a separate algorithm variable in each of the log entries, to specify a checksum algorithm to be used separately for each said log entry.

30. (Original) A storage server as recited in claim 24, further comprising:  
means for maintaining an entry count in the log to indicate the number of log entries in the log; and  
means for using the checksum of one of the log entries to determine whether the entry count is corrupted.
31. (Original) A storage server as recited in claim 24, wherein the storage appliance is a network appliance.
32. (New) A method for operating a network-accessible data storage server, comprising:  
receiving a plurality of storage requests from at least one client;  
preparing a plurality of log entries, each log entry of the plurality of log entries corresponding to one storage request of the plurality of storage requests, and each log entry including a checksum of the log entry; and  
storing the plurality of log entries in a non-volatile random access memory (“NVRAM”).
33. (New) The method of claim 32, further comprising:  
preparing a log header containing a count of the plurality of log entries;  
and  
storing the log header in the NVRAM.
34. (New) The method of claim 33, further comprising:  
storing a monotonically increasing serial number in each of the plurality of log entries;  
identifying a minimum serial number of the plurality of log entries as a start serial number;

verifying a log entry with a serial number equal to a sum of the start serial number and the count of the plurality of log entries; and

verifying a log entry with a serial number equal to a sum of the start serial number and the count of the plurality of log entries and one.

35. (New) The method of claim 33 further comprising:

computing a checksum of the log header and storing the checksum with the log header in the NVRAM.